



# CRUD DAN FORM

Created by [Aris Prawisudatama](#)

## HELLO THERE

I am

## ARIS PRAWISUDATAMA

A Software Developer.

# GITHUB REPOSITORY

[https://github.com/soedomoto/usm\\_tis21213p](https://github.com/soedomoto/usm_tis21213p)

# GITHUB REPOSITORY

[https://github.com/soedomoto/usm\\_tis21213p](https://github.com/soedomoto/usm_tis21213p)

# BACKGROUND

Hampir semua aplikasi, termasuk sistem informasi,  
memiliki penyimpanan data

Agar data tidak hilang (persistent)

Harus ada mekanisme manajemen data

CRUD - Create, Read, Update, Delete

Tidak boleh langsung diubah didatabase

Integritas data, Hak Akses

# BACKGROUND

Hampir semua aplikasi, termasuk sistem informasi,  
memiliki penyimpanan data

Agar data tidak hilang (persistent)

Harus ada mekanisme manajemen data

CRUD - Create, Read, Update, Delete

Tidak boleh langsung diubah didatabase

Integritas data, Hak Akses

# BACKGROUND

Hampir semua aplikasi, termasuk sistem informasi

memiliki penyimpanan data

Agar data tidak hilang (persistent)

Harus ada mekanisme manajemen data

CRUD - Create, Read, Update, Delete

Tidak boleh langsung diubah didatabase

Integritas data, Hak Akses

# BACKGROUND

Hampir semua aplikasi, termasuk sistem informasi

memiliki penyimpanan data

Agar data tidak hilang (persistent)

Harus ada mekanisme manajemen data

CRUD - Create, Read, Update, Delete

Tidak boleh langsung diubah didatabase

Integritas data, Hak Akses

# BACKGROUND

Hampir semua aplikasi, termasuk sistem informasi

memiliki penyimpanan data

Agar data tidak hilang (persistent)

Harus ada mekanisme manajemen data

CRUD - Create, Read, Update, Delete

Tidak boleh langsung diubah didatabase

Integritas data, Hak Akses

# BACKGROUND

Hampir semua aplikasi, termasuk sistem informasi

memiliki penyimpanan data

Agar data tidak hilang (persistent)

Harus ada mekanisme manajemen data

CRUD - Create, Read, Update, Delete

Tidak boleh langsung diubah didatabase

Integritas data, Hak Akses

# BACKGROUND

Hampir semua aplikasi, termasuk sistem informasi

memiliki penyimpanan data

Agar data tidak hilang (persistent)

Harus ada mekanisme manajemen data

CRUD - Create, Read, Update, Delete

Tidak boleh langsung diubah di database

Integritas data, Hak Akses

## PREREQUISITES

Apache/Nginx + PHP + MySQL (XAMPP, etc)

Editor dengan Code Highlighting (VSCode, Sublime,  
Notepad++)

# PREREQUISITES

Apache/Nginx + PHP + MySQL (XAMPP, etc)

Editor dengan Code Highlighting (VSCode, Sublime, Notepad++)

# PREREQUISITES

Apache/Nginx + PHP + MySQL (XAMPP, etc)

Editor dengan Code Highlighting (VSCode, Sublime, Notepad++)

# CRUD - CREATE, READ, UPDATE, DELETE

Create: Menambahkan record pada database

Read: Membaca satu atau lebih record pada database

Update: Mengubah record pada database

Delete: Menghapus record pada database

Hard delete, Soft delete

# CRUD - CREATE, READ, UPDATE, DELETE

Create: Menambahkan record pada database

Read: Membaca satu atau lebih record pada database

Update: Mengubah record pada database

Delete: Menghapus record pada database

Hard delete, Soft delete



# CRUD - CREATE, READ, UPDATE, DELETE

Create: Menambahkan record pada database

Read: Membaca satu atau lebih record pada database

Update: Mengubah record pada database

Delete: Menghapus record pada database

Hard delete, Soft delete

# CRUD - CREATE, READ, UPDATE, DELETE

Create: Menambahkan record pada database

Read: Membaca satu atau lebih record pada database

Update: Mengubah record pada database

Delete: Menghapus record pada database

Hard delete, Soft delete

# CRUD - CREATE, READ, UPDATE, DELETE

Create: Menambahkan record pada database

Read: Membaca satu atau lebih record pada database

Update: Mengubah record pada database

Delete: Menghapus record pada database

Hard delete, Soft delete

# CRUD - CREATE, READ, UPDATE, DELETE

Create: Menambahkan record pada database

Read: Membaca satu atau lebih record pada database

Update: Mengubah record pada database

Delete: Menghapus record pada database

Hard delete, Soft delete

# SINTAKS SQL

## Create

```
1 INSERT INTO `mahasiswa` (`nim`, `nama`) VALUES
2   ('11111', 'Ahmad'),
3   ('11112', 'Ade');
```

## Read

```
1 SELECT * FROM `mahasiswa`;
```

# SINTAKS SQL

## Update

```
1 UPDATE `mahasiswa` SET `nama` = 'Ahmad Subagja'
2 WHERE `mahasiswa`.`nim` = '11111';
```

## Delete

```
1 DELETE FROM mahasiswa WHERE `mahasiswa`.`nim` = '11111';
```

# HTTP METHOD

## GET

Address bar browser

Biasa dipake untuk request data

## POST

Form

Biasa dipake untuk mengirim data

# HTTP METHOD

## GET

Address bar browser

Biasa dipake untuk request data

## POST

Form

Biasa dipake untuk mengirim data

# HTTP METHOD

GET

Address bar browser

Biasa dipake untuk request data

POST

Form

Biasa dipake untuk mengirim data

# HTTP METHOD

GET

Address bar browser

Biasa dipake untuk request data

POST

Form

Biasa dipake untuk mengirim data

# HTTP METHOD

GET

Address bar browser

Biasa dipake untuk request data

POST

Form

Biasa dipake untuk mengirim data

# HTTP METHOD

GET

Address bar browser

Biasa dipake untuk request data

POST

Form

Biasa dipake untuk mengirim data

# HTTP METHOD

GET

Address bar browser

Biasa dipake untuk request data

POST

Form

Biasa dipake untuk mengirim data

## HTTP METHOD: JARANG DIGUNAKAN

DELETE

Biasa dipake untuk menghapus data via API

HEAD

Sama dengan GET, tetapi tidak memiliki body

OPTIONS

Biasanya untuk mendapatkan permission, seperti CORS

PUT

Biasanya untuk mengupdate data via API

# HTTP METHOD: JARANG DIGUNAKAN

## DELETE

Biasa dipake untuk menghapus data via API

## HEAD

Sama dengan GET, tetapi tidak memiliki body

## OPTIONS

Biasanya untuk mendapatkan permission, seperti CORS

## PUT

Biasanya untuk mengupdate data via API

# HTTP METHOD: JARANG DIGUNAKAN

## DELETE

Biasa dipake untuk menghapus data via API

## HEAD

Sama dengan GET, tetapi tidak memiliki body

## OPTIONS

Biasanya untuk mendapatkan permission, seperti CORS

## PUT

Biasanya untuk mengupdate data via API



# HTTP METHOD: JARANG DIGUNAKAN

DELETE

Biasa dipake untuk menghapus data via API

HEAD

Sama dengan GET, tetapi tidak memiliki body

OPTIONS

Biasanya untuk mendapatkan permission, seperti CORS

PUT

Biasanya untuk mengupdate data via API

# HTTP METHOD: JARANG DIGUNAKAN

DELETE

Biasa dipake untuk menghapus data via API

HEAD

Sama dengan GET, tetapi tidak memiliki body

OPTIONS

Biasanya untuk mendapatkan permission, seperti CORS

PUT

Biasanya untuk mengupdate data via API

# HTTP METHOD: JARANG DIGUNAKAN

DELETE

Biasa dipake untuk menghapus data via API

HEAD

Sama dengan GET, tetapi tidak memiliki body

OPTIONS

Biasanya untuk mendapatkan permission, seperti CORS

PUT

Biasanya untuk mengupdate data via API

# HTTP METHOD: JARANG DIGUNAKAN

DELETE

Biasa dipake untuk menghapus data via API

HEAD

Sama dengan GET, tetapi tidak memiliki body

OPTIONS

Biasanya untuk mendapatkan permission, seperti CORS

PUT

Biasanya untuk mengupdate data via API

# HTTP METHOD: JARANG DIGUNAKAN

DELETE

Biasa dipake untuk menghapus data via API

HEAD

Sama dengan GET, tetapi tidak memiliki body

OPTIONS

Biasanya untuk mendapatkan permission, seperti CORS

PUT

Biasanya untuk mengupdate data via API

# HTTP METHOD: JARANG DIGUNAKAN

DELETE

Biasa dipake untuk menghapus data via API

HEAD

Sama dengan GET, tetapi tidak memiliki body

OPTIONS

Biasanya untuk mendapatkan permission, seperti CORS

PUT

Biasanya untuk mengupdate data via API

# LETS PRACTICE

## PDO - PHP DATA OBJECT

PHP Extension yang menyediakan interface generic untuk mengakses database

Analog dengan ODBC dan JDBC

Membutuhkan driver sesuai target database

Pros: Interface tetap sama, meskipun menggunakan database yang berbeda

# PDO - PHP DATA OBJECT

PHP Extension yang menyediakan interface generic untuk  
mengakses database

Analog dengan ODBC dan JDBC

Membutuhkan driver sesuai target database

Pros: Interface tetap sama, meskipun menggunakan database yang berbeda

# PDO - PHP DATA OBJECT

PHP Extension yang menyediakan interface generic untuk  
mengakses database

Analog dengan ODBC dan JDBC

Membutuhkan driver sesuai target database

Pros: Interface tetap sama, meskipun menggunakan database yang berbeda

# PDO - PHP DATA OBJECT

PHP Extension yang menyediakan interface generic untuk

mengakses database

Analog dengan ODBC dan JDBC

Membutuhkan driver sesuai target database

Pros: Interface tetap sama, meskipun menggunakan database yang berbeda

# PDO - PHP DATA OBJECT

PHP Extension yang menyediakan interface generic untuk

mengakses database

Analog dengan ODBC dan JDBC

Membutuhkan driver sesuai target database

Interface tetap sama, meskipun menggunakan database yang berbeda

# PDO - PHP DATA OBJECT

```
1 $host = 'db';
2 $db   = 'usm';
3 $user = 'user';
4 $pass = 'pass';
5 $port = "3306";
6 $charset = 'utf8mb4';
7
8 $options = [
9     \PDO::ATTR_ERRMODE           => \PDO::ERRMODE_EXCEPTION,
10    \PDO::ATTR_DEFAULT_FETCH_MODE => \PDO::FETCH_ASSOC,
11    \PDO::ATTR_EMULATE_PREPARES   => false,
12 ];
13
14 $dsn = "mysql:host=$host;dbname=$db;charset=$charset;port=$port";
15 $pdo = new \PDO($dsn, $user, $pass, $options);
```

## PDO - PREPARED VS UNPREPARED

Unprepared

Query langsung dieksekusi

Tidak dapat menggunakan placeholder: ? atau :field

```
1 $stmt = $pdo->query("SELECT * FROM mahasiswa WHERE nim = '11'");
2 $rows = $stmt->fetchAll();
```

# PDO - PREPARED VS UNPREPARED

## Unprepared

Query langsung dieksekusi

Tidak dapat menggunakan placeholder: ? atau :field

```
1 $stmt = $pdo->query("SELECT * FROM mahasiswa WHERE nim = '1111111111'");
2 $rows = $stmt->fetchAll();
```

# PDO - PREPARED VS UNPREPARED

## Unprepared

Query langsung dieksekusi

Tidak dapat menggunakan placeholder: ? atau :field

```
1 $stmt = $pdo->query("SELECT * FROM mahasiswa WHERE nim = '1111111111'");
2 $rows = $stmt->fetchAll();
```



# PDO - PREPARED VS UNPREPARED

## Unprepared

Query langsung dieksekusi

Tidak dapat menggunakan placeholder: ? atau :field

```
1 $stmt = $pdo->query("SELECT * FROM mahasiswa WHERE nim = '11111'");
2 $rows = $stmt->fetchAll();
```

# PDO - PREPARED VS UNPREPARED

## Prepared

Menjadi template. Tidak langsung dieksekusi

Dapat menggunakan placeholder: ? atau :field

Dapat dieksekusi berkali-kali dengan mengganti parameter

```
1 $stmt = $pdo->prepare("SELECT * FROM mahasiswa WHERE nim = :nim");
2
3 $stmt->execute([':nim' => '11111']);
4 $rows = $stmt->fetchAll();
5 var_dump($rows);
6
7 $stmt->execute([':nim' => '11112']);
8 $rows = $stmt->fetchAll();
9 var_dump($rows);
```

# PDO - PREPARED VS UNPREPARED

## Prepared

Menjadi template. Tidak langsung dieksekusi

Dapat menggunakan placeholder: ? atau :field

Dapat dieksekusi berkali-kali dengan mengganti parameter

```
1 $stmt = $pdo->prepare("SELECT * FROM mahasiswa WHERE nim = :");
2
3 $stmt->execute([':nim' => '11111']);
4 $rows = $stmt->fetchAll();
5 var_dump($rows);
6
7 $stmt->execute([':nim' => '11112']);
8 $rows = $stmt->fetchAll();
9 var_dump($rows);
```

# PDO - PREPARED VS UNPREPARED

## Prepared

Menjadi template. Tidak langsung dieksekusi

Dapat menggunakan placeholder: ? atau :field

Dapat dieksekusi berkali-kali dengan mengganti parameter

```
1 $stmt = $pdo->prepare("SELECT * FROM mahasiswa WHERE nim = :");
2
3 $stmt->execute([':nim' => '11111']);
4 $rows = $stmt->fetchAll();
5 var_dump($rows);
6
7 $stmt->execute([':nim' => '11112']);
8 $rows = $stmt->fetchAll();
9 var_dump($rows);
```

# PDO - PREPARED VS UNPREPARED

## Prepared

Menjadi template. Tidak langsung dieksekusi  
Dapat menggunakan placeholder: ? atau :field  
Dapat dieksekusi berkali-kali dengan mengganti parameter

```
1 $stmt = $pdo->prepare("SELECT * FROM mahasiswa WHERE nim = :");
2
3 $stmt->execute([':nim' => '11111']);
4 $rows = $stmt->fetchAll();
5 var_dump($rows);
6
7 $stmt->execute([':nim' => '11112']);
8 $rows = $stmt->fetchAll();
9 var_dump($rows);
```

# PDO - PREPARED VS UNPREPARED

## Prepared

Menjadi template. Tidak langsung dieksekusi  
Dapat menggunakan placeholder: ? atau :field  
Dapat dieksekusi berkali-kali dengan mengganti parameter

```
1 $stmt = $pdo->prepare("SELECT * FROM mahasiswa WHERE nim = :");
2
3 $stmt->execute([':nim' => '11111']);
4 $rows = $stmt->fetchAll();
5 var_dump($rows);
6
7 $stmt->execute([':nim' => '11112']);
8 $rows = $stmt->fetchAll();
9 var_dump($rows);
```

# LIST MAHASISWA

Menggunakan GET

Tidak ada data yang dikirim

Dapat memiliki parameter sebagai filter

# LIST MAHASISWA

Menggunakan GET

Tidak ada data yang dikirim

Dapat memiliki parameter sebagai filter

# LIST MAHASISWA

Menggunakan GET

Tidak ada data yang dikirim

Dapat memiliki parameter sebagai filter

# LIST MAHASISWA

Menggunakan GET

Tidak ada data yang dikirim

Dapat memiliki parameter sebagai filter

# LIST MAHASISWA

```
1 $stmt = $pdo->prepare("SELECT * FROM mahasiswa");
2 $stmt->execute();
3
4 $rows = $stmt->fetchAll();
```

```
1 <!--?php foreach ($rows as $r) { ?-->
2
3     <!--?php echo $r['nim']; ?-->
4     <!--?php echo $r['nama']; ?-->
5
6     <a href="./ubah?nim=?php echo $r['nim']; ?>" class='
7         Edit
8     </a>
9     <a href="./hapus?nim=?php echo $r['nim']; ?>" class=
10     Hapus
11     </a>
12
13
14 <!--?php } ?-->
```

## TAMBAH MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat data yang dikirim

# TAMBAH MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat data yang dikirim

# TAMBAH MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat data yang dikirim

# TAMBAH MAHASISWA

```
1 <form class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">
2   <div class="mb-4">
3     <label class="block text-gray-700 text-sm font-bold mb-2">
4       NIM
5     </label>
6     <input class="shadow appearance-none border rounded w-100" type="text">
7   </div>
8   ...
9   <div class="flex items-center justify-between">
10    <a class="inline-block align-baseline font-bold text-sm" href="#"></a>
11    Kembali
12  </a>
13    <button class="bg-sky-500 hover:bg-sky-700 px-5 py-2 text-white">
14      Simpan
15    </button>
```

# TAMBAH MAHASISWA

```
1 if ($_POST) {
2   $nim = $_POST['nim'];
3   $nama = $_POST['nama'];
4
5   $stmt = $pdo->prepare("INSERT INTO mahasiswa (nim, nama) \
6   $stmt->execute(['nim' => $nim, 'nama' => $nama]);
7
8   $lastInsertedId = $pdo->lastInsertId();
9 }
```



# TAMBAH MAHASISWA

```
1 <!--?php if (isset($lastInsertedId)) { ?-->
2 <div class="bg-teal-100 border-t-4 border-teal-500 rounded
3 <div class="flex">
4 <div class="py-1"><svg class="fill-current h-6 w-6 te
5 <path d="M2.93 17.07A10 10 0 1 1 17.07 2.93 10 10
6 </svg></div>
7 <div>
8 <p class="font-bold">Tambah Mahasiswa</p>
9 <p class="text-sm">Tambah mahasiswa berhasil.</p>
10 </div>
11 </div>
12 </div>
13 <!--?php } ?-->
```

# UBAH MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat parameter yang dikonsumsi

Terdapat data yang dikirim

# UBAH MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat parameter yang dikonsumsi

Terdapat data yang dikirim

# UBAH MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat parameter yang dikonsumsi

Terdapat data yang dikirim

# UBAH MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat parameter yang dikonsumsi  
Terdapat data yang dikirim

# UBAH MAHASISWA

```
1 $stmt = $pdo->prepare("SELECT * FROM mahasiswa WHERE nim = :  
2 $stmt->execute(['nim' => $_GET['nim']] );  
3 $row = $stmt->fetch();
```

# UBAH MAHASISWA

```
1 <form class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4"
2   <div class="mb-4">
3     <label class="block text-gray-700 text-sm font-bold mb-2">
4       NIM
5     </label>
6     <input class="shadow appearance-none border rounded w-100" type="text">
7   </div>
8   ...
9   <div class="flex items-center justify-between">
10     <a class="inline-block align-baseline font-bold text-sm" href="#"></a>
11     Kembali
12   </a>
13     <button class="bg-sky-500 hover:bg-sky-700 px-5 py-2 text-white">
14     Simpan
15   </button>
```

# UBAH MAHASISWA

```
1 if ($_POST) {
2   $nim = $_GET['nim'];
3   $nama = $_POST['nama'];
4
5   $stmt = $pdo->prepare("UPDATE mahasiswa SET nama = :nama WHERE nim = :nim");
6   $isUpdated = $stmt->execute(['nim' => $nim, 'nama' => $nama]);
7 }
```

# UBAH MAHASISWA

```
1 <!--?php if (isset($isUpdated) && $isUpdated == true) { ?--  
2 <div class="bg-teal-100 border-t-4 border-teal-500 rounded  
3 <div class="flex">  
4 <div class="py-1"><svg class="fill-current h-6 w-6 te  
5 <path d="M2.93 17.07A10 10 0 1 1 17.07 2.93 10 10  
6 </svg></div>  
7 <div>  
8 <p class="font-bold">Ubah Mahasiswa</p>  
9 <p class="text-sm">Ubah mahasiswa berhasil.</p>  
10 </div>  
11 </div>  
12 </div>  
13 <!--?php } ?-->
```

# HAPUS MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat parameter yang dikonsumsi

Terdapat data yang dikirim

# HAPUS MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat parameter yang dikonsumsi

Terdapat data yang dikirim

# HAPUS MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat parameter yang dikonsumsi

Terdapat data yang dikirim

# HAPUS MAHASISWA

Menggunakan kombinasi GET dan POST

Terdapat parameter yang dikonsumsi  
Terdapat data yang dikirim

# HAPUS MAHASISWA

```
1 $stmt = $pdo->prepare("SELECT * FROM mahasiswa WHERE nim = :  
2 $stmt->execute(['nim' => $_GET['nim']] );  
3 $row = $stmt->fetch();
```

# HAPUS MAHASISWA

```
1 <form class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4"
2   <input name="nim" type="hidden" value="<?php echo $row['r
3   <div class="flex items-center justify-between">
4     <a class="inline-block align-baseline font-bold text-sm
5       Batal
6     </a>
7     <button class="bg-red-500 hover:bg-red-700 px-5 py-2 te
8       Ya, Hapus
9     </button>
10  </div>
11 </form>
```

# HAPUS MAHASISWA

```
1 if ($_POST) {
2   $nim = $_POST['nim'];
3   $stmt = $pdo->prepare("DELETE FROM mahasiswa WHERE nim = :
4   $isDeleted = $stmt->execute(['nim' => $nim]);
5 }
```



# HAPUS MAHASISWA

```
1 <!--?php if (isset($isDeleted) && $isDeleted == true) { ?--
2 <div class="bg-teal-100 border-t-4 border-teal-500 rounded
3 <div class="flex">
4 <div class="py-1"><svg class="fill-current h-6 w-6 te
5 <path d="M2.93 17.07A10 10 0 1 1 17.07 2.93 10 10
6 </svg></div>
7 <div>
8 <p class="font-bold">Hapus Mahasiswa</p>
9 <p class="text-sm">Hapus mahasiswa berhasil.</p>
10 <a class="inline-block align-baseline font-bold tex
11 <div>
12 <div>
13 <div>
14 <div>
15 </div>
```

## TRIVIA

### ARIA - ACCESSIBLE RICH INTERNET APPLICATIONS

Digunakan oleh assistive technology (AT), seperti Braille atau TTS  
Untuk menambahkan informasi dari informasi normal di HTML

## TRIVIA

# ARIA - ACCESSIBLE RICH INTERNET APPLICATIONS

nakan oleh assistive technology (AT), seperti Braille atau  
Untuk menambahkan informasi dari informasi normal di HTML

## TRIVIA

# ARIA - ACCESSIBLE RICH INTERNET APPLICATIONS

nakan oleh assistive technology (AT), seperti Braille atau  
uk menambahkan informasi dari informasi normal di HT

**SEKIAN**